

We kunnen zelden alle gewenste data bewaren in aparte variabelen (vb wanneer je een programma moet maken dat het gemiddelde berekent van 100 getallen). Aangezien het aantal variabelen dat moet worden bijgehouden afhankelijk is van verschillende factoren is het nodig om constructies te gebruiken die deze variërende factor kunnen opvangen. Arrays zijn de meest primitieve vorm van massa containers die we in C# zullen gebruiken. Collections zijn hier een uitbreiding op en bieden extra functionaliteit die het werken met deze gegevensopslag makkelijker maakt. Arrays zijn echter de basis van collecties en vele eigenschappen (zoals het indexing mechanisme) vinden we terug in de geavanceerde Collections.

De “tekortkomingen” van Arrays werden weggewerkt in de collections objecten (ArrayList) en nog meer in de generische vorm hiervan (vb de populaire generische List<Type>). Deze collections laten ons toe om gemakkelijk een aantal objecten bij te houden zonder, tijdens de ontwikkeling van ons programma, de grootte te moeten aangeven. Collections zijn dus dynamische Arrays. Verder gaan collections ons ook verschillende functies aanbieden die het werken met deze structuren gemakkelijker zal maken.

Oefening 1

Maak een PCShop die de gebruiker de mogelijkheid biedt om:

- Een PC toe te voegen (na het ingeven van alle vereiste informatie)
- Een lijst van alle PC's op te vragen
- Een lijst van alle PC's die voldoen aan een minimum aantal Gb geheugen

PC

1. Properties:

- Code [string]: elke PC heeft zijn eigen code
- Geheugen [int]: grootte geheugen (Gb)

2. Constructoren

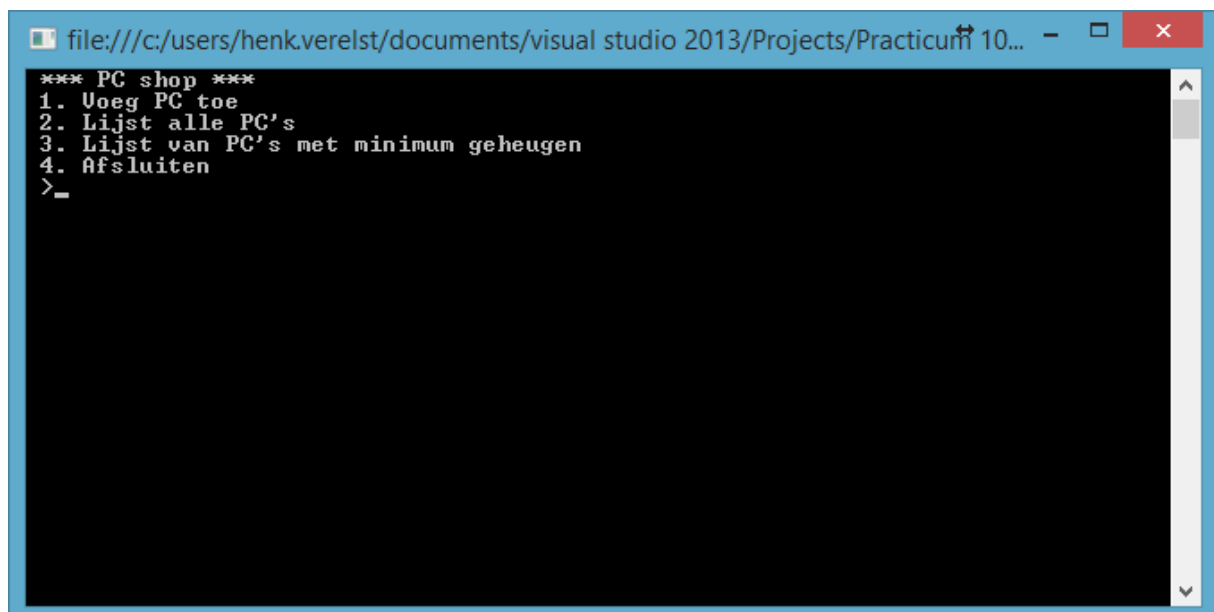
- Default constructor: deze constructor zal alle properties initialiseren op 0 en blanco afhankelijk van het type
- Constructor met parameters: deze constructor zal voor elke property een waarde voorzien waarmee deze eerste dan ook geïnitieerd worden

Je kan de onderstaande code hiervoor gebruiken om de PC lijst te voorzien van enkele modellen:

```
store.Add(new PC("DF6", 16));
store.Add(new PC("ZER6D23", 8));
store.Add(new PC("6SDS", 32));
store.Add(new PC("WXVC5", 12));
store.Add(new PC("YI5", 8));
store.Add(new PC("G56GT", 16));
store.Add(new PC("SQF65", 10));
store.Add(new PC("897DS", 8));
store.Add(new PC("VML56DJ", 8));
store.Add(new PC("4SLK5", 16));
```

Programma screenshots:

Hoofdmenu:



```
file:///c:/users/henk.verelst/documents/visual studio 2013/Projects/Practicum 10... - x
*** PC shop ***
1. Voeg PC toe
2. Lijst alle PC's
3. Lijst van PC's met minimum geheugen
4. Afsluiten
>_
```

1. Voeg PC toe

```
file:///c:/users/henk.verelst/documents/visual studio 2013/Projects/Practicum 10... - [X]
*** PC shop ***
1. Voeg PC toe
2. Lijst alle PC's
3. Lijst van PC's met minimum geheugen
4. Afsluiten
>1

Geef de gegevens van de nieuwe PC in
Model   : DF9032
Geheugen : 24
De PC werd toegevoegd

Druk op een toets om verder te gaan
```

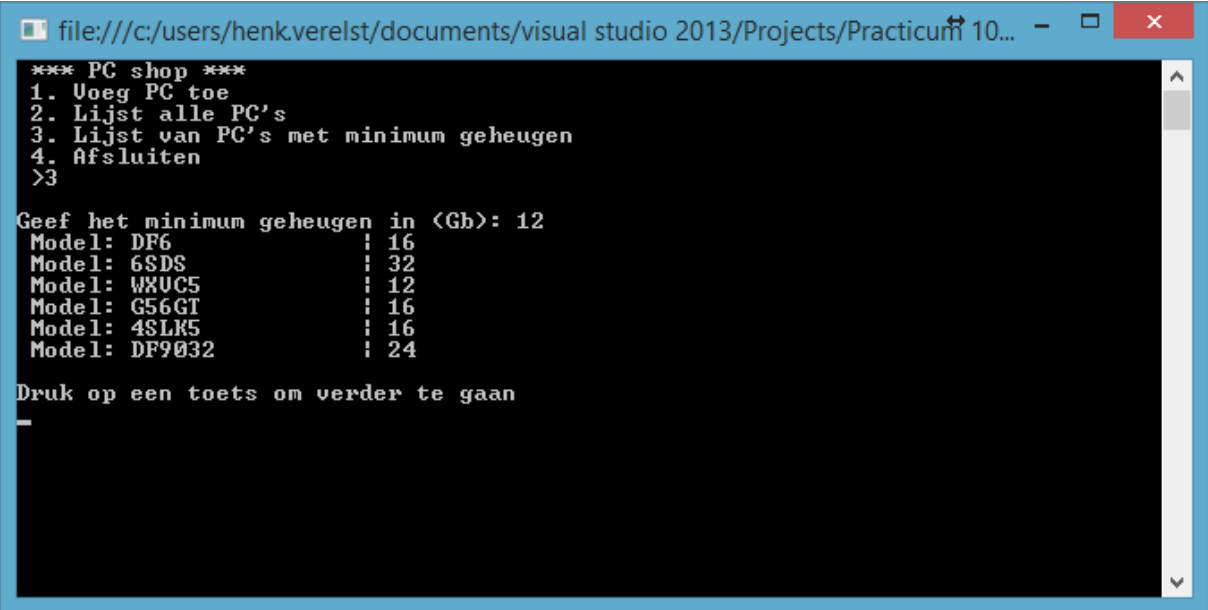
2. Lijst alle PC's

```
file:///c:/users/henk.verelst/documents/visual studio 2013/Projects/Practicum 10... - [X]
*** PC shop ***
1. Voeg PC toe
2. Lijst alle PC's
3. Lijst van PC's met minimum geheugen
4. Afsluiten
>2

Winkel overzicht
Model: DF6           : Gb:16
Model: ZER6D23       : Gb:8
Model: 6SDS          : Gb:32
Model: WXUC5         : Gb:12
Model: YI5           : Gb:8
Model: G56GT         : Gb:16
Model: SQF65         : Gb:10
Model: 897DS         : Gb:8
Model: UML56DJ       : Gb:8
Model: 4SLK5         : Gb:16
Model: DF9032        : Gb:24

Druk op een toets om verder te gaan
```

3. Lijst van PC's met minimum geheugen



```
file:///c:/users/henk.verelst/documents/visual studio 2013/Projects/Practicum 10... - [X]
*** PC shop ***
1. Voeg PC toe
2. Lijst alle PC's
3. Lijst van PC's met minimum geheugen
4. Afsluiten
>3

Geef het minimum geheugen in (Gb): 12
Model: DF6      : 16
Model: 6SDS     : 32
Model: WXUC5    : 12
Model: G56GT    : 16
Model: 4SLK5    : 16
Model: DF9032   : 24

Druk op een toets om verder te gaan
-
```